Approximate Message Authentication and Biometric Entity Authentication*

G. Di Crescenzo[1] R. Graveman[2] R. Ge[3] G. Arce[3]

[1] Telcordia Technologies, Piscataway, NJ
E-mail: giovanni@research.telcordia.com

[2] Work done while at Telcordia Technologies.
E-mail: rfg@acm.org

[3] University of Delaware, Newark, DE
E-mail: {ge,arce}@ece.udel.edu

Abstract. Approximate Message Authentication Code (AMAC) is a recently introduced cryptographic primitive with several applications in the areas of cryptography and coding theory. Briefly speaking, AMACs represent a way to provide data authentication that is tolerant to acceptable modifications of the original message. Although constructs had been proposed for this primitive, no security analysis or even modeling had been done.

In this paper we propose a rigorous model for the design and security analysis of AMACs and show how to transform any ordinary MAC into an AMAC. Our constructions have short output, leading to efficient storage or communication complexity.

AMACs is a useful primitive with several applications of different nature. A major one, that we study in this paper, is that of entity authentication via biometric techniques or passwords over noisy channels. We present a formal model for the design and analysis of biometric entity authentication schemes and show simple and natural constructions of such schemes using any AMAC.

1 Introduction

The rise of financial crimes such as identity theft (recent surveys show there are currently 7-10 million victims per year) and check fraud (more than 500 million checks are forged annually with losses totaling more than 10 Billion dollars in the United States alone) is challenging financial institutions to meeting

high security levels of entity authentication and data integrity. Passwords are a good start to secure access to their systems but, when used alone, don't seem enough to provide the security and convenience level for identification needed by financial organizations. (Passwords can be compromised, stolen, shared, or just forgotten.) Biometrics, on the other hand, are based on a user's unique biological characteristics, and can be an effective additional solution to the entity authentication problem for financial systems. One challenge in implementing biometric authentication is, however, the reliability of the system with respect to errors in repeated measurements of the same biometric data, such as fingerprints, voice messages, or iris scans.

In this paper we put forward a formal model for the study of approximate data authentication schemes, that are tolerant with respect to errors in the data, and therefore are suitable for the verification of biometric data in entity authentication schemes. We also present an efficient construction of approximate data authentication, leading to efficient constructions for various types of biometric entity authentication schemes.

Data Authentication. A fundamental cryptographic primitive is that of Message Authentication Codes (MAC), namely, methods for convincing a recipient of a message that the received data is the same that originated from the sender. MACs are extremely important in today's design of secure systems since they reveal to be useful both as atomic components of more complex cryptographic systems and as themselves alone, to guarantee integrity of stored and transmitted data. Traditional message authentication schemes create a hard authenticator, where modifying a single message bit would result in a modification of about half the authentication tag. These MACs fit those applications where the security requirement asks to reject any message that has been altered to the minimal extent. In many other applications, such as those concerning biometric data, there may be certain modifications to the message that may be acceptable to sender and receiver, such as errors in reading biometric data or in communicating passwords through very noisy channels. This new scenario, not captured by the traditional notion of MACs, motivated the introduction and study in [5] of a new cryptographic primitive, a variant of MACs, which was called Approximate Message Authentication Code (AMAC); namely, methods that propagate "acceptable" modifications to the message to "recognizable" modifications in the authentication tag, and still retain their security against other, unacceptable modifications. Examples of the applicability of AMACs include: message authentication in highly-noisy or highly-adversarial communication channels, as in mobile ad hoc networks; simultaneous authentication of sets of semantically equivalent messages; and, of specific interest in this paper, entity authentication through inherently noisy data, such as biometrics or passwords over noisy channels.

Our contributions. If, on one hand, after investigations in [5], the intended notion of AMAC was precisely formulated, on the other hand, a rigorous model for the security study of AMACs was not. Therefore, a problem implicitly left open by [5] was that of establishing such a model. In this paper we propose a

rigorous model for analyzing approximation in message authentication. It turns out that the issue of approximation has to be considered in both the correctness property (if Alice and Bob share a key and follow the protocol, then Bob accepts the message) and the security property (no efficient adversary not knowing the shared key and mounting a chosen message attack can make Bob accept a new message). Our notions of approximate correctness and approximate security use as a starting point the previously proposed notions for conventional MACs and address one difficulty encountered in both allowing acceptable modifications to the message and achieving a meaningful security notion. In addition, we formulate a preimage-resistance and a public-verifiability requirement that make these AMACs especially applicable to two variants of biometric entity authentication problems.

Our main AMAC construction uses finite pseudo-random functions and universal one-way hash functions to transform any MAC into an AMAC that satisfies all mentioned requirements. One step in this transformation solves the technical problem of constructing a probabilistic universal one-way hash function with distance-preserving properties.

We then show how to apply this construction (and, in fact, any AMAC construction) to obtain simple and efficient biometric entity authentication schemes in both a closed-network and an open-network setting.

Formal proofs of our theorems are either sketched or omitted due to lack of space.

Related work. References in conventional Message Authentication Codes are discussed in Section 2. Universal one-way hash function were introduced in [14] and are being often applied in cryptographic constructions. Related work to AMACs includes work from a few different research literatures.

There is a large literature that investigates biometric techniques without addressing security properties (see, e.g. [7] and references therein). Security and privacy issues in biometrics have been independently recognized and advocated by many researchers (see, e.g., [3, 15, 16]).

A second literature (related to information and coding theory) investigates techniques for authentication of noisy multimedia messages (see, e.g., [12, 13] and references therein). All these constructs either ignore security issues or treat them according to information theoretic models. Typically, constructions of the latter type have a natural adaptation to the symmetric MAC setting but all constructions we found, after this adaptation, fail to satisfy the MAC requirement of security under chosen message attack (and therefore the analogue AMAC requirement). Some work (e.g, [11]) uses digital signatures as atomic components but they result in constructions that are not preimage-resistant, according to our Definition 2, and therefore cannot be applied to give a satisfactory solution to our biometric authentication problem.

A third literature investigates coding and combinatorial techniques for error tolerance in biometrics (see, e.g., [9, 8]), as well as privacy amplification from reconciliation. Recently, [4, 2] considered the problem of generating strongly random keys from biometric data. These constructions can be used to define a solution

to the problem of biometric entity authentication. In particular, the solution in [4] suffices for single-use biometric entity authentication, and the results in [2], in addition to show that the solution in [4] can be broken if the same biometric is used multiple times, imply a solution to a variant of (interactive) biometric entity authentication. These papers address primitives and notions (fuzzy commitments, fuzzy extractors, etc.) unaddressed by ours and viceversa.

We stress that all this work did not even imply a formal definition of AMACs.

2 Definitions and Preliminaries

In this section we present our novel definition of Approximate MACs. In the rest of the paper we will assume familiarity with definitions of pairwise-independent hash functions and of cryptographic primitives used in the paper, such as universal one-way hash functions, (conventional) MACs, symmetric encryption schemes and finite pseudo-random functions.

Approximation in MACs. We introduce formal definitions for approximate MACs, using as a starting point the above definitions for ordinary MACs. Informally, one would like an approximate MAC to be tolerant to "acceptable" modifications to the original message. Less informally, we will define approximate versions of the same properties as an ordinary MAC, where the approximation is measured according to some polynomial-time computable distance function on the message space. For the correctness property, the notion of a modification being acceptable is formalized by requiring an authentication tag computed for some message m, to be verified as correct even for messages having up to a given distance from m. We note that this property might not be compatible with the property of security against chosen message attack, for the following reason. The latter property makes an adversary unable to produce a valid pair of message and authentication tag, for a new message, for which he hasn't seen an authentication tag so far; the former property, instead, requires the receiver himself to be able to do so for some messages, that is, for messages having a certain distance from the original message obtained from the sender. In order to avoid this apparent definitional contradiction, we define a chosen message attack to be successful if the valid pair of message and authentication tag produced by the adversary contains a message which has a larger distance from all messages for which he has seen an authentication tag during his chosen message attack. Therefore, we even define the security property for MACs in some approximate sense.

We now proceed more formally.

Definition 1. Let M denote the message space and let d_m be a polynomial-time computable distance function over M. An *approximately correct and approximately secure message authentication code for distance function* d_m (briefly, d_m-ac-as-MAC) is a triple (Kg,Tag,Verify), where the polynomial-time algorithms Kg, Tag, Verify satisfy the same syntax as in the definition of MACs, except that now these three algorithms are parameterized by function d_m. Moreover, we define the following two requirements.

1. (p, δ)-*Approximate Correctness*: after k is generated using Kg, if tag is generated using algorithm Tag on input message m and key k, then, with probability at least p, algorithm Verify, on input k, m', tag, outputs: yes, if $d_m(m, m') \leq \delta$.
2. $(d_m, \gamma, t, q, \epsilon)$-*Approximate Security*: Let k be generated using Kg; for any algorithm Adv running in time at most t, if Adv queries algorithm $\mathrm{Tag}(k, \cdot)$ with adaptively chosen messages, thus obtaining pairs $(m_1, t_1), \ldots, (m_q, t_q)$, and then returns a pair (m, t), the probability that $\mathrm{Verify}(k, m, t) = yes$ and $d_m(m, m_i) \geq \gamma$ for $i = 1, \ldots, q$, is at most ϵ.

Note that (t, q, ϵ)-secure MAC schemes are (p, δ)-approximately correct and $(d_m, \gamma, t, q, \epsilon)$-approximately secure MAC schemes for $p = 1$, $\delta = 0$, $\gamma = 1$, and d_m equal to the Hamming distance. In the sequel, we will omit d_m in the term d_m-ac-as-MAC when clear from the context, or directly abbreviate the term d_m-ac-as-MAC as AMAC.

Two additional properties of AMACs. A first additional property that we can require from AMACs is that of preimage-resistance. Informally, we require that the tagging algorithm, if viewed as a function on the message space, is hard to invert, no matter what is the distribution on the message space. (Later, while showing the applications of AMACs to biometric entity authentication, this property will be useful in proving that the entity authentication scheme obtained is secure against adversaries that can gain access to the AMAC output from the biometric storage file.)

Definition 2. The d_m-ac-as-MAC (Kg,Tag,Verify) is (t, ϵ)-*preimage-resistant* if the following holds. Let k be generated using Kg; for any algorithm Adv running in time at most t, if Adv queries algorithm $\mathrm{Tag}(k, \cdot)$ with adaptively chosen messages, thus obtaining pairs $(m_1, t_1), \ldots, (m_q, t_q)$, and then returns a message m_0, and is given a value $tag = \mathrm{Tag}(k, m)$, the probability that $Adv(tag)$ returns m' such that $\mathrm{Verify}(k, m', tag) = 1$ and $d_m(m', m_i) \geq \gamma$ for $i = 0, 1, \ldots, q$, is at most ϵ.

We note that essentially all conventional MAC constructions in the literature would satisfy an analogue preimage-resistance requirement. However it is easy to transform a MAC into one that is not preimage-resistant and for some applications like biometric identification, it can be desirable to require that the MAC used is preimage-resistant (or otherwise an accidental loss of the MAC output could reveal a password or some biometric data to an adversary).

A second additional property that we can require from AMACs is that of tag public verifiability. Informally, we require that, given two tags obtained from two different messages, it is possible to efficiently verify that the two messages have small distance, without using any secret key. (Later, while showing the applications of AMACs to biometric entity authentication, this property will be useful in obtaining a network entity authentication scheme, where the server does not need to run the AMAC to verify tag correctness.)

Definition 3. The d_m-ac-as-MAC (Kg,Tag,Verify) has (d_m, δ, γ)-*publicly verifiable tags* if there exists an efficient algorithm PubVerify such that the following holds. Let k be generated using Kg and let tag_i =Tag(k, m_i), for $i = 1, 2$. Then PubVerify(tag_1, tag_2)=1 if $d(m_1, m_2) \leq \delta$ and 0 if $d(m_1, m_2) \geq \gamma$.

Previous work on AMACs. Previously to this work, variations of the same approximate MAC contruction had been proposed and investigated in [5, 17]. Informally, the tagging algorithm in these constructions uses operations such as xoring the message with a pseudo-random string of the same length, computing a pseudo-random permutation of the message, and returning majority values of subsets of message bits. The security of these constructions was not analyzed in a cryptographic model.

Simple attempts towards AMAC constructions. First of all, we remark that several simple constructions using arbitrary error correcting codes and ordinary MACs fail in satisfying even the approximate correctness and security requirements of AMACs. These include techniques such as interpreting the input message as a codeword, and using a conventional MAC to authenticate its decoding (here, the property of approximate correctness fails). Other techniques that also fail are similar uses of fuzzy commitments from [9], fuzzy sketches from [4] and reusable fuzzy extractors from [2]. We note however that there are a few simple constructions that meet the approximate correctness and security requirements of AMACs but don't meet the preimage-resistance or the efficiency or the tag public verifiability requirement. The simplest we found goes as follows. Let us denote as (K,T,V) a conventional MAC scheme. The tagging algorithm, on input key k and message m, returns $tag = m \,|\, \mathrm{T}(k, m)$. The verifying algorithm, on input k, m', tag, sets $tag = t1 \,|\, t2$ and returns 1 if and only if $d(t1, m') \leq \delta$ and $V(k, t1, t2) = 1$, where d is the distance function. The scheme satisfies the approximate correctness and security, and the tag public verifiability requirements; however, note that the tag of this scheme contains the message itself and therefore the scheme is neither preimage-resistant nor efficient.

3 Our AMAC Constructions

In this section we present two constructions of approximately-correct and secure MAC with respect to the Hamming distance. The first construction is based on systematic error correcting codes and is preimage-resistant but does not have publicly verifiable tags. The second construction, the main one in the paper, transforms a MAC into an AMAC that is additionally both preimage-resistant and tag publicly verifiable.

3.1 A Preimage-Resistant AMAC Construction

A basic construction of an ac-as-MAC for the Hamming distance function can be obtained by using any conventional MAC scheme, any symmetric encryption

scheme, and any appropriate systematic error correcting code. The construction satisfies approximate correctness with $p = 1$, approximate security under minimal assumptions, and preimage resistance.

Formal description. Let us denote by (K_m,T,V) a conventional MAC scheme, and by (K_e,E,D) a conventional symmetric encryption scheme. Also, by (SEnc,SDec) we denote a systematic error-correcting code (that is, on input m, SEnc(m) = c, where $c = m|pc$, and pc are parity check bits), such that the decoding algorithm perfectly recovers the message if at most δ errors happened or returns failure symbol $\perp$ otherwise (note that this latter condition is without loss of generality as any error correcting code can be simply transformed into one that satisfies it).

Instructions for Kg: generate a uniformly distributed k-bit key K

Input to Tag: two k-bit keys K_a, K_e, an n-bit message M, parameters p, δ, γ.

Instructions for Tag:

1. Set $c = Enc(M)$ and write c as $c = M|pc$
2. Set $subtag = T_{K_a}(M)$ and $epc = E(K_e, pc)$
3. **Return:** $tag = epc|subtag$ and halt.

Input to Verify: parameters p, δ, γ, two k-bit keys K_a, K_e, an n-bit message M' and a string tag

Instructions for Verify:

1. Write tag as $tag = epc|subtag$
2. Let $pc = D(K_e, epc)$ and $m' = Dec(M'|pc)$
3. If $m' = \perp$ then **Return:** 0
4. If $V(K_a, m', subtag) = 1$ then **Return:** 1 else **Return:** 0.

We can prove the following

Theorem 1. Let d_m denote the Hamming distance, let n be the length of the input message for (Kg,Tag,Verify) and let (SEnc,SDec) a systematic error-correcting code that corrects up to δ errors and returns $\perp$ if more than δ errors happened. If (K_m,T,V) is a (t, q, ϵ)-secure MAC then (Kg,Tag,Verify) is a (p, δ)-approximately correct and $(d_m, \gamma, t', q', \epsilon')$-approximately secure MAC for $p = 1$, $\gamma = \delta + 1$, $t' = t - O(q \cdot |c|)$, $q' = q$, $\epsilon' = \epsilon$, where $|c|$ is the length of the output returned by Enc on inputs of size n. Moreover, if (K_m,T,V) is preimage-resistant and (K_e,E,D) is a secure symmetric encryption scheme then (Kg,Tag,Verify) is preimage-resistant.

The above theorem already provides ac-as-MACs with some useful properties, such as approximate correctness, approximate security and preimage-resistance. However, we note two facts that make this scheme not a definitely satisfactory solution: first, its tag length depends on the performance of the systematic code used, and can thus be significantly longer than regular MACs even for moderately

large values of the parameter δ; second, this scheme does not satisfy the tag public verifiability property. As we will see in Section 4, the latter is essential in order to construct a main application of AMACs: a network biometric entity authentication scheme. The scheme in Section 3.2 satisfies both efficiency of tag length (for any value of δ) and the tag public verifiability property.

3.2 Our Main AMAC Construction

Informal description. We explain the ideas behind this scheme in two steps. First, we explain how to use a probabilistic TCR hash function to guarantee that outputs from this hash function will have some additional distance-preserving properties. Second, we show how we can use such probabilistic TCR hash function to construct an approximately correct and secure MAC.

We achieve a combination of distance-preserving properties and target collision resistance by making a TCR hash function probabilistic, and using the following technique. First, the message bits are randomly permuted and then the resulting message is written as the concatenation of several equal-size blocks. Here, the size of each block could be the fixed constant size (e.g., 512 bits) of the input to compression functions (e.g., SHA) that are used as atomic components of practical constructions of TCR hash functions. Now multiple hashes are computed, each being obtained using the TCR hash function, using as input the concatenation of a different and small enough subset of the input blocks. Here, the choice of each subset is done at random, and specifically, using the output of a random pairwise-independent hash function on input the message. Furthermore, each subset has the same size, depending on the length of the input and on the desired distance-preserving properties. The basic idea so far is that by changing the content of some blocks of the message, we only change a small fraction of the inputs of the atomic hashes and therefore only a small fraction of the outputs of those hashes will change.

Given this 'probabilistic TCR hash function', the tagging and verifying algorithm can be described as follows.

The tagging algorithm, on input a random key and a message, uses another value, which can be implemented as a counter incremented after each application (or a random value chosen independently at each application). Then the algorithm computes the output of the finite pseudo-random function on input such value and divides this output in two parts: the first part is a random key for the TCR hash function and the second part is a sequence of pseudo-random bits that can be used as randomness for the above described probabilistic TCR hash function. Now, the tagging algorithm can run the latter function to compute multiple hashes of the message. The tag returned is then the input to the finite pseudo-random function and the hashes.

The construction of the verifying algorithm is necessarily differently from the usual approach for exactly correct and secure MACs (where the verifying algorithm runs the tagging algorithm on input the received message and checks that its output is equal to the received tag), as this algorithm needs to accept

the same tag for multiple messages. Specifically, on input the tag returned by the tagging algorithm, the verifying algorithm generates a key and pseudo-random bits for the probabilistic TCR hash function exactly as the tagging algorithm does and computes the hashes of the received message. Finally, the verifying algorithm checks that the received and the computed sequences of hashes only differ in a small enough number of positions.

Formal description. Let k be a security parameter, t be an approximation parameter, and c be a block size constant. We denote by $PIH = \{pih \,|\, pih : \{0,1\}^n \to \{0,1\}^n\}$ a set of pairwise independent hash functions over $\{0,1\}^m$, by $TCRH = \{tcrh_K : K \in \{0,1\}^k\}$ a finite TCR hash function, and by $F = \{f_K : K \in \{0,1\}^k\}$ a finite pseudo-random function. We now present our construction of an approximately-secure and approximately-correct MAC, which we denote as (Kg,Tag,Verify).

Instructions for Kg: generate a uniformly distributed k-bit key K

Input to Tag: a k-bit key K, an n-bit message M, parameters p, δ, γ, a block size 1^c and a counter ct.

Instructions for Tag:

- Set $x_1 = n/2\delta$ and $x_2 = 10\log(1/(1-p))$
- Set $(u|\pi|pih) = f_K(ct)$, where $u \in \{0,1\}^k$, π is a permutation of $\{0,1\}^n$ and $pih \in PIH$
- Write $\pi(M)$ as $M_1|\cdots|M_{\lceil n/c \rceil}$, where $|M_i| = c$ for $i = 1, \ldots, \lceil n/c \rceil$
- Use $pih(M)$ as randomness to randomly choose x_1-size subsets $S_1, \ldots, S_{x_2}$ of $\{1, \ldots, \lceil n/c \rceil\}$
- For $i = 1, \ldots, x_2$,
  let $N_i = M_{i_1}|\cdots|M_{i_{x_1}}$, where $S_i = \{i_1, \ldots, i_{x_1}\}$
  let $sh_i = tcrh_u(N_i)$
- Let $subtag = sh_1|\cdots|sh_{x_2}$
- **Return:** $tag = ct|subtag$.
- Set $ct = ct + 1$ and halt.

Input to Verify: parameters δ, γ, a block size 1^c, a k-bit key K, an n-bit message M' and a string tag

Instructions for Verify:

- Write tag as $ct|u|sh_1|\cdots|sh_{x_2}$
- Set $x_1 = n/2\delta$ and $x_2 = 10\log(1/(1-p))$
- Set $(u|\pi|pih) = f_K(ct)$, where $u \in \{0,1\}^k$, π is a permutation of $\{0,1\}^n$ and $pih \in PIH$
- Write $\pi(M')$ as $M'_1|\cdots|M'_{\lceil n/c \rceil}$, where $|M'_i| = c$ for $i = 1, \ldots, \lceil n/c \rceil$
- Use $pih(M')$ to randomly select x_1-size subsets $S'_1, \ldots, S'_{x_2}$ of $\{1, \ldots, \lceil n/c \rceil\}$
- For $i = 1, \ldots, x_2$,
  let $N'_i = M'_{i_1}|\cdots|M'_{i_{x_1}}$, where $S'_i = \{i_1, \ldots, i_{x_1}\}$
  let $sh'_i = tcrh_u(N'_i)$

– Check that $sh'_i = sh_i$, for at least αx_2 of the values of $i \in \{1, \ldots, x_2\}$, for $\alpha = 1 - 1/2\sqrt{e} - 1/2e$.
– **Return:** 1 if all verifications were successful and 0 otherwise.

The above construction satisfies the following

Theorem 2. Assume that F is a (t_F, q_F, ϵ_F)-secure pseudo-random function and H is a (t_H, q_H, ϵ_H)-target collision resistant hash function. Then (Kg,Tag,Verify) is a (p, δ)-approximately correct and $(d_m, \gamma, t_A, q_A, \epsilon_A)$-approximately secure MAC, where

- d_m is the Hamming distance
- $\gamma = 2\delta$
- $\epsilon_A \leq \epsilon_F + \epsilon_H \cdot q_A + 1 - p$
- $q_A = q_F \geq 1$ and $q_H = 10 \log(1/(1-p))$
- $t_A = \min(t_{A,1}, t_{A,2})$
- $t_{A,1} = t_F - O(q_A(n(\log n + \log(1/(1-p))) + \log(1/(1-p)) + time(h_u; nc/2\delta))$
- $t_{A,2} = t_H - O(n(\log n + \log(1/(1-p))) + time(f_K; |ct|))$

and n is the length of the message, c is a block size constant, ct is the counter input to algorithm Tag, and $time(g; x)$ denotes the time required to compute function g on inputs of size x.

Performance. We analyze the main performance parameter of interest; that is, the communication complexity of our scheme (Kg,Tag,Verify). We see that the length of the returned tag is $x_2 \cdot c$, where $x_2 = 10 \log(1/(1-p))$, and c is the length of the output of the TCR hash function. We note that c is constant with respect to n, and acceptable settings of parameter p can lie anywhere in the range $[1 - 1/2^{(\log n)^{1+\epsilon}}, 1]$, for any constant $\epsilon > 0$, and where n is the length of the message input to the scheme. Therefore the length of the tag returned by the scheme can be as small as $10c(\log n)^{1+\epsilon}$; most importantly, this holds for *any* value of parameter δ. The tag length remains much shorter than the message even for much larger settings of p; for instance, if $p = 1 - 2^{-\sqrt{n}}$, the tag length becomes $O(\sqrt{n})$.

3.3 Properties of our Main Construction

We divide the proof of Theorem 2 in two parts: first we prove the property of approximate correctness and then the property of approximate security.

Approximate correctness. Assume $d_m(M, M') \leq \delta$. Moreover, assume that f_K is a random function. Then, for $i = 1, \ldots, x_2$, define random variable X_i as equal to 1 if $sh_i \neq sh'_i$ or 0 otherwise. Furthermore, we denote by N_i and $M_{i1}, \ldots, M_{i_{x_1}}$ (resp., N'_i and $M'_{i1}, \ldots, M'_{i_{x_1}}$) the values used in the 5th step of algorithm **Tag** on input M (resp., M'). Then it holds that

$$\begin{aligned} a &= \text{Prob}\,[\,X_i = 1\,] \\ &\leq 1 - \text{Prob}\,[\,N_i = N'_i\,] \\ &\leq 1 - \text{Prob}\,\big[\,M_{i_1} = M'_{i_1}\,\big]^{n/2\delta} \\ &\leq 1 - ((n-\delta)/n)^{n/2\delta} = 1 - (1 - \delta/n)^{n/2\delta} \;\leq\; 1 - 1/\sqrt{e}, \end{aligned}$$

where the first inequality follows from the definition of X_i, the second inequality follows from the definition of N_i and N'_i, and the third inequality follows from the uniform and independent choice of subsets S_i and S'_i and therefore of the blocks M_i and M'_i among all blocks in $\pi(M)$ and $\pi(M')$, respectively. We set $\alpha - a = (e-1)/4e^2$. Since $X_1, \ldots, X_{x_2}$ are independent and identically distributed, we can apply a Chernoff bound and obtain that

$$\text{Prob}\left[\sum_{i=1}^{x_2} X_i \; < \; \alpha x_2\right] \; \leq \; e^{-2(\alpha-a)^2 x_2} \; \leq \; 1-p,$$

which implies that algorithm Verify returns 1 with probability at least p. Note that the assumption that f_K is a random function can be removed by only subtracting a negligible factor to p, as otherwise the pseudorandomness f_K can be contradicted.

Approximate security. We assume that the requirement of $(d_m, \gamma, t, q, \epsilon)$-approximate security is not satisfied and reach some contradiction. The proof for this (only sketched here) requires the definition of four probability experiments that slightly differ from each other.

Experiment 1 is precisely the experiment in the definition of approximate security. We denote by p_1 the probability that experiment 1 is successful; our original assumption implies that $p_1 > \epsilon$.

Experiment 2 differs from experiment 1 only in that Adv queries a finite random function r rather than a finite pseudo-random function Tag. Denoting as p_2 the probability that experiment 2 is successful, we can prove that $p_2 - p_1 \leq \epsilon_F$, or otherwise Adv can be used to violate the assumption that F is a (t_F, q_F, ϵ_F)-secure pseudo-random function.

Experiment 3 is a particular case of experiment 2; specifically, it is successful when experiment 2 is and the following happens: the adversary returns a tag with the same counter as in a tag previously returned by the oracle and, moreover, it produces at least one hash equal to one hash previously seen during the chosen message attack. Since the adversary returns a tag with the same counter as in a tag previously returned by the oracle, it also uses the same key for the target collision resistant hash function. Furthermore, since it produces at least one hash equal to one hash previously seen under the same hash function, it violates the security of the hash function. We denote as p_3 the probability that experiment 3 is successful, and obtain that $p_3 \leq \epsilon_H \cdot q_A$, or otherwise Adv can be used to violate the assumption that H is a (t_H, q_H, ϵ_H)-target collision resistant hash function,

Experiment 4 is a particular case of experiment 2, but it considers the case complementary to the case in experiment 3. Specifically, the adversary produces no hash equal to any hash previously seen during the chosen message attack or does not copy any of the previously seen counters. Since this experiment is a particular case of experiment 2 and considers the case complementary to the case in experiment 3, we obtain that $p_2 \leq p_3 + p_4$. We denote as p_4 the probability that experiment 4 is successful, and observe that this experiment is successful only if the adversary is lucky in obtaining a sufficiently large number of the

subsets S'_i that contain no block where M_i and M'_i differ. We observe that such subsets are generated according to a distribution uniform and independent in both cases we consider in this experiment, as we now explain. In the first case, a different counter ct is returned by Adv (that is, $ct \neq ct^i$, for $i = 1, \ldots, q_A$) and therefore the subsets are generated using $pih(m)$ as randomness, where pih is part of the value $r(ct)$, the latter being independently distributed from $r(ct^i)$, for $i = 1, \ldots, q_A$. In the second case, a counter is copied (that is, $ct = ct^j$, for exactly one $j \in \{1, \ldots, q_A\}$) but the subsets are generated using the subsets are generated using $pih(m)$ as randomness, where pih is part of the value $r(ct) = r(ct^j)$, but $pih(m)$ is uniformly and independently distributed from $pih(m_j)$, as $m \neq m_j$ and pih is pairwise-independent. Given that the subsets are uniformly and independently distributed, using a Chernoff bound and the same analysis as in the proof of the approximate correctness property, we can show that the probability that Adv can make algorithm Verify return 1 is at most $1 - p$. Therefore we obtain that $p_4 \leq 1 - p$.

We conclude the analysis by using the obtained inequalities: $p_1 - p_2 \leq \epsilon_F$, $p_2 \leq p_3 + p_4$, $p_3 \leq \epsilon_H \cdot q_A$, and $p_4 \leq 1 - p$; and therefore obtaining that $\epsilon_A \leq p_1 \leq \epsilon_F + \epsilon_H \cdot q_A + 1 - p$.

4 Biometric Entity Authentication

We present a model for the design and analysis of biometric entity authentication (BEA) schemes, and show that two simple constructions based on AMACs can be proved secure in our model under standard assumptions on cryptographic tools and biometric distribution.

Our model. There is a server S and several users $U_1, \ldots, U_m$, where the server has a biometric storage file bsf and each user U_i is associated with a biometric b_i, a reader R_i and a computing unit CU_i, for $i = 1, \ldots, m$. We define a (non-interactive) BEA scheme between user U_i and S as the following two-phase protocol. The first phase is an *initialization phase* during which user U_i and S agree on various parameters and shared keys and S stores some information on bsf. The second phase is the *authentication phase*, including the following steps. First, user U_i inputs her biometric b_i to the reader R_i, which extracts some feature information $fb_{i,t}$ (this may be a sketched version of the original biometric b_i) and returns a measurement $mb_{i,t}$, where t here represents the time when R_i is executed. (Specifically, the reader may return a different value $mb_{i,t}$ for each different time t, on input the same b_i.) Then the computing unit CU_i, on input $mb_{i,t}$ sends an authenticating value $ab_{i,t}$ to the server, that, using information stored during the initialization phase, decides whether to accept $ab_{i,t}$ as a valid value for user U_i or not.

The *correctness* requirement for a BEA scheme states that the following happens with high probability: after the initialization phase is executed between $U_i(b_i)$ and S, if, for some t, $mb_{i,t} = R_i(b_i)$, and $ab_{i,t} = CU_i(mb_{i,t})$ then S accepts pair $(U_i, ab_{i,t})$.

An *adversary Adv* tries to attack a BEA scheme by entering a biometric b_j into a reader R_i, and, before doing that, can have access to several and different resources, according to which parties it can corrupt (i.e., noone; users U_j, for $j \neq i$; server S; etc.), and which communication lines or storage data he has access to (i.e., none; the communication lines containing any among $mb_{i,t}, ab_{i,t}$; the biometric storage file bsf; the server's secret keys; user U_i's secret keys, etc.). The *security* requirement for a BEA scheme states that after the initialization phase is executed between $U_i(b_i)$ and S, for $i = 1, \ldots, m$, the probability that an efficient adversary *Adv* can input his biometric b_j into a reader R_i, for $i \neq j$, and make S accept the resulting pair $(U_i, ab^j_{i,t})$, is negligible.

We are now ready to show two simple BEA constructions given any AMAC scheme with certain properties (in fact, to achieve security against certain realistic adversaries, our constructions may even assume a AMAC secure against a weaker adversary than the one in Definition 1). The first construction is for *local* BEA; that is, the adversary has no access to the measurements $mb_{i,t}$ and the user can send them in the clear to the server. Local BEA is comparable, in terms of both functionality and security, to well-known password-based authentication schemes in non-open networks. The second construction is for *network* BEA; that is, the message sent from a user to a server during the authentication phase can travel through an open network. Network BEA should be contrasted, in terms of both functionality and security, to password-based authentication schemes in open networks; in particular, we will show that our scheme does not require a user to send over an open network (not even in encrypted form) a reading of her biometric. Both constructions necessarily make an assumption on the distribution of biometric that we now describe.

A basic assumptions on biometrics. Biometric entity authentication (in any model) inherently relies on the assumption that there exist a distance function d, appropriate parameters $\delta < \gamma$, and an efficiently computable measurement M of biometrics such that: (1) for each individual with a biometric b with feature information $fb(t)$ at time t, and for any times t_1, t_2, it holds that $d(\mathrm{M}(fb(t_1)),\mathrm{M}(fb(t_2))) \leq \delta$; (2) for any two individuals with biometrics b_1, b_2, with feature information $fb_1(t), fb_2(t)$ at time t, respectively, and for any times t_1, t_2, it holds that $d(\mathrm{M}(fb(t_1)),\mathrm{M}(fb(t_2))) \geq \gamma$. We refer to this as the *Biometric Distribution Assumption* (BD Assumption).

A construction for local BEA. Informally, the first construction consists of the user sending the reading of her biometric to the server, that checks it against the previously stored AMAC tag of a reading done at initialization phase. More formally, let (Kg,Tag,Verify) denote an AMAC scheme. Then the BEA scheme lAmacBEA goes as follows. During the initialization phase, user U_i sends ab_{i,t_0} to the server S, that stores $tag_0 = \mathrm{Tag}(k, ab_{i,t_0})$ in the bsf file. During the authentication phase, at time t_1, user U_i inputs b_i into the reader R_i, that returns mb_{i,t_1}; the latter is input to CU_i that returns $ab_{i,t_1} = mb_{i,t_1}$; finally, pair (U_i, ab_{i,t_1}) is sent to S. On input pair (U_i, ab_{i,t_1}), server S computes $\mathrm{Verify}(k, ab_{i,t_1}, tag_0)$ and accepts U_i if and only if it is equal to 1.

We can prove the following

Theorem 3. Under the BD assumption, if (Kg,Tag,Verify) is an AMAC scheme then the construction lAmacBEA is a BEA scheme satisfying the above correctness and security requirement against efficient adversaries that can corrupt up to all users U_j but one. Furthermore, if scheme (Kg,Tag,Verify) is preimage-resistant then the construction lAmacBEA satisfies security against efficient adversaries that additionally have access to the biometric storage file bsf.

A construction for network BEA. Informally, the second construction modifies the first construction by having the user compute the AMAC tag over the reading of her biometric; the AMAC tag is then sent to the server that can check it (without need for the AMAC key) against the previously stored AMAC tag of a reading done at initialization phase. More formally, let (Kg,Tag,Verify) denote an AMAC scheme with publicly verifiable tags. Also, we assume for simplicity that the channel between each user and the server is properly secured (using standard encryption and authentication techniques), and so is the biometric storage file (using standard encryption techniques), Then the BEA scheme nAmacBEA goes as follows. During the initialization phase, user U_i inputs her biometric b_i into reader R_i, that returns mb_{i,t_0}; the latter is input to CU_i that returns and sends ab_{i,t_0} =AMAC(k, mb_{i,t_0}) to S; finally, S stores ab_{i,t_0} into bsf. The authentication phase is very similar to the identification phase; specifically, user U_i computes ab_{i,t_1} in the same way, and pair (U_i, ab_{i,t_1}) is sent to S, that computes PubVerify(ab_{i,t_0}, ab_{i,t_1}) and accepts U_i if and only if it is equal to 1.

We can prove the following

Theorem 4. Under the BD assumption, if (Kg,Tag,Verify) is an AMAC scheme with publicly verifiable tags, then the construction nAmacBEA is a BEA scheme satisfying the above correctness and security requirement against efficient adversaries that can corrupt up to all users U_j but one and have access to the communication lines containing $mb_{i,t}, ab_{i,t}$ and the server's secret keys. Furthermore, if scheme (Kg,Tag,Verify) is preimage-resistant then the construction nAmacBEA satisfies security against efficient adversaries that additionally have access to the biometric storage file bsf.

We note that the first AMAC construction in Section 3 is preimage-resistant and therefore suffices for the AMAC scheme required by Theorem 3. Furthermore, the second AMAC construction in Section 3 is both preimage-resistant (this follows by using the definition of universal one-way functions) and has publicly verifiable tags (this can be noted by inspection of the algorithm Verify), and therefore can be used to construct the AMAC scheme required by Theorem 4.

References

1. J. Black, S. Halevi, H. Krawczyk, T. Krovetz, and P. Rogaway, *UMAC: Fast and Secure Message Authentication*, Proc. of CRYPTO '99.
2. X. Boyen, *Reusable Cryptographic Fuzzy Extractors,* Proc. of 11th ACM CCS 04.
3. G. Davida, Y. Frankel, and B. Matt, *On Enabling Secure Application through Off-Line Biometric Identification,* IEEE 1998 Symposium on Research in Security and Privacy
4. Y. Dodis, L. Reyzin, and A. Smith, *Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data,* Proc. of Eurocrypt 2004
5. R. F. Graveman, L. Xie, and G. R. Arce, *Approximate Message Authentication Codes,* submission to the IEEE Transactions on Image Processing, 2000.
6. P. Indyk, R. Motwani, P. Raghavan, and S. Vempala, *Locality-Preserving Hashing in Multidimensional Spaces,* Proc. of STOC 97
7. A. Jain, R. Bolle, and S. Pankanti, eds. BIOMETRICS: PERSONAL IDENTIFICATION IN A NETWORKED SOCIETY, Kluwer Academic Publishers, 1999.
8. A. Juels and M. Sudan, *A Fuzzy Vault Scheme,* Proc. of IEEE ISIT 2002.
9. A. Juels and M. Wattenberg, *A Fuzzy Commitment Scheme,* Proc. of ACM CCS 1999
10. N. Linial and O. Sasson, *Non-Expansive Hashing,* Proc. of STOC 96
11. E. Martinian, *Authenticating Multimedia in the Presence of Noise,* Master Thesis.
12. E. Martinian, B. Chen and G. Wornell, *Information Theoretic Approach to the Authentication of Multimedia,* Proc. of SPIE Conference on Electronic Imaging, 2001
13. E. Martinian, B. Chen and G. Wornell, *On Authentication With Distortion Constraints,* Proc. of IEEE International Symposium on Information Theory, 2001
14. M. Naor and M. Yung, *Universal one-way hash functions and their cryptographic applications,* Proc. of STOC 89.
15. S. Prabhakar, S. Pankanti, and A. Jain, *Biometric Recognition: Security and Privacy Concerns,* IEEE Security and Privacy, March 2003.
16. B. Schneier, *Inside Risks: The Uses and Abuses of Biometrics,* Communications of the ACM, vol. 42, no. 8, pp. 136, Aug. 1999.
17. L. Xie, G. R. Arce, and R. F. Graveman, *Approximate Image Message Authentication Codes,* IEEE Transactions on Multimedia, vol. 3, June 2001.